Computational Models of Evolutionary Learning

Philip G.K. Reiser[1]

Abstract: This paper surveys progress in the simulation of evolutionary learning and characterises some of the mechanisms that have been abstracted. In nature, mechanisms have evolved that permit increasingly rapid and complex adaptations to the environment. We observe that evolutionary learning systems are adopting these mechanisms to tackle harder and harder learning problems. Extrapolating this trend, we indicate an interesting new direction for future work on evolutionary learning systems.

Keywords: Darwinian evolution, machine learning, genetic algorithms, multi-agent learning.

1 Introduction

Nature has been the source of inspiration for many of the paradigms in machine learning. In this paper, we focus specifically on natural mechanisms and processes viewed from an evolutionary perspective. Evolutionary computing, which concerns models of learning based on natural evolution, continues to attract much interest in the machine learning community. The literature, going back over 30 years, describes numerous learning algorithms based on the simulation of evolution, some of which have been successfully applied to real-world problems.

However, there are still many difficulties that inhibit still wider applicability of these algorithms. For example, even when applied to relatively simple problem domains, computational cost is high and other approaches frequently solve the same problem far more efficiently. Two principle reasons for this failing are (1) many candidate solutions are examined that cannot be interpreted as valid solutions; and (2) it is difficult to aid evolutionary learning systems by supplying them with *a priori* knowledge. Typically, the algorithm begins *tabula rasa*, even when relevant domain knowledge is readily available.

One of the aims of this paper is to examine how evolutionary learning systems address the problem of learning with respect to (1) efficiently search the space of solutions, (2) exploiting domain knowledge, and (3) producing results that are intelligible. This paper begins by examining a small subset of the processes that account for evolution and proceeds to review how these processes have been modelled in learning algorithms. Finally, we consider the significance of these processes and indicate directions for further work.

[1]Centre for Intelligent Systems, University of Wales, Aberystwyth, Wales, UK. e-mail: pgr94@aber.ac.uk

2 Evolution at a Glance

Before proceeding to consider existing models of evolutionary learning, we begin by examining a few selected mechanisms and processes found in natural evolution. Our aim is to identify and isolate a few evolutionary 'milestones' and assess their contribution to evolutionary learning. These include genetic mutation and recombination, nervous systems, symbolic reasoning and structured communication through language. This list does not seek to be complete and our treatment can necessarily only be at a superficial level—there are numerous books and journals that address thses subjects individually. We examine these processes in a chronological approach to mirror their evolution.

2.1 Nature's Programs

The properties of an organism, such as its morphology, physiology, and behaviour are determined by the interpretation of an internal plan. This plan, the *genotype*, comprises of information structures that have two levels of interpretation. At one level, information structures may be interpreted as data that can be manipulated and transformed. At the other level, information structures may be interpreted as instructions that encode the various characteristics of an organism, or its *phenotype*. This dual level of interpretation shares a striking similarity to the instructions or *program* supplied to a digital computer. Consequently, evolution and machine learning tackle very similar problems: the transformation of programs. By studying the processes involved in natural evolution we seek to find mechanisms that will inspire the design of machine learning algorithms.

2.2 Necessary Processes for Evolution

It has been argued, [8], that evolution can be accounted for merely by a small number of stochastic processes. They are reproduction, mutation, competition, and selection.

> Neo-Darwinism asserts that the history of the vast majority of life is fully accounted for by only a very few statistical processes acting on and within populations and species [18, p.39]. These processes are reproduction, mutation, competition, and selection. Reproduction is an obvious property of all life. But similarly as obvious, mutation is guaranteed in any system that continuously reproduces itself in a positively entropic universe. Competition and selection become the inescapable consequences of any expanding population constrained to a finite arena. Evolution is then the result of these fundamental interacting stochastic processes as they act on populations, generation after generation [22], [35, p.25] and others. ([8, p.37]).

A consequence of this point of view is that these four processes represent the foundation from which all the complexity of life has originated.

3 Asexual Reproduction

These four processes are sufficient to allow the genotype to be transformed such that the phenotype is highly adapted to the organism's environment. However, the efficiency of adaptation at this basic level is very poor. Purely random mutation turns out to be a double-edged sword. As the mutation rate increases, the possible rate of adaptation also increases, but simultaneously, there is a higher risk of destructive mutations occurring.

So any organism that exhibits a slightly better means of adaptation than purely random mutation, will propagate rapidly. Indeed, the processes reported in the biological literature are far more complex than purely random mutation. For example, there is increasing evidence to suggest that there exist mechanisms encoded in the genotype that are able regulate the rate of mutation on parts of the genotype, [25, 31, 30].

3.1 Simulation of Asexual Reproduction

Evolutionary Programming, [11, 12], is a class of evolutionary algorithms that is based upon reproduction, selection, competition, and mutation. Mutation is an operation that causes perturbations in a continuous range of behavioural diversity while ensuring a behavioural link between each parent and its offspring, [10]. Such approaches have been successfully applied to some difficult real-world problems, (e.g. the travelling salesman problem, [9]).

4 Genetic Recombination

In asexual reproduction, the only changes to the genotype come about by chance mutations. Offspring are genetically similar, if not identical, to the parent. With the evolution of genetic recombination, offspring genotypes stem not only from a small number of changes to the genotype, but are also composed of parts of the genotype from *both* parents. Therefore, the offspring is practically never identical to either parent.

Consequently, not only are the changes more frequent, but also the number of possible genotypes that can be constructed may be significantly larger than those stemming from a small number of mutations. The offspring genotypes can be considerable more diverse, and hence there is greater scope for adaptation to the environment.

So, while sexual reproduction has disadvantages associated with it (e.g. finding a suitable mate), there is also a significant benefit from the point of view of adaptation.

4.1 Genetic Recombination Methods

There are several classes of evolutionary algorithm that are based on genetic recombination. Genetic recombination, or *crossover*, is used to attempt to propagate good partial solutions, or *building blocks*, to subsequent generations. As described by Goldberg's building block hypothesis: "instead of building high-performance strings by trying every conceivable combination, [genetic recombination] constructs better and better strings from the best partial solutions of past samplings", [14, p.41]. Unfortunately, there is a caveat. The success of crossover relies on the representation allowing good partial solutions to be formed. We briefly describe several basic models.

The Genetic Algorithm, [20, 14], is distinguished by a fixed-length binary string representation and crossover operator that does not affect the length of the string and may generate illegal representations. An example is illustrated in Figure 1(a). Some successful applications are described in [15].

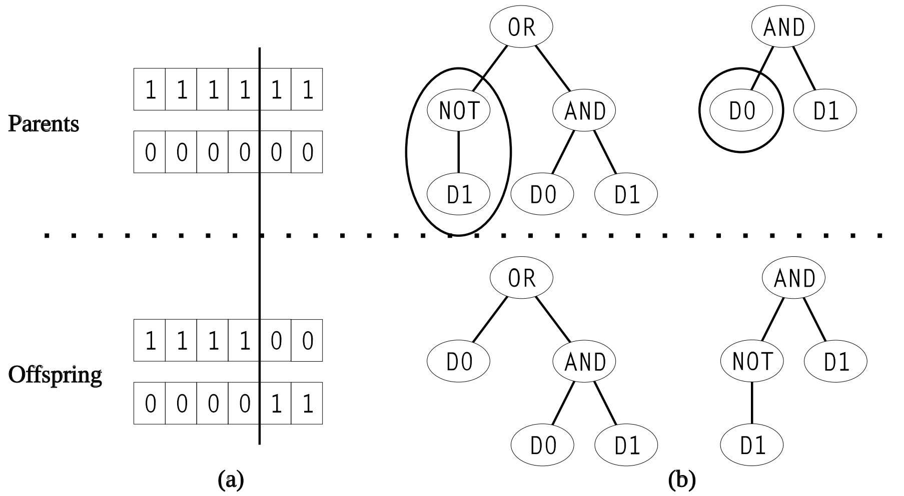

Figure 1: (a) crossover of fixed-length binary strings in genetic algorithms; (b) crossover of trees of Lisp S-expressions in genetic programming.

In Genetic Programming, [24], the structures undergoing adaptation are general, hierarchical computer programs of dynamically varying shape and size. These are expressed as trees of Lisp S-expressions. New programs are produced by moving branches from one tree and inserting them into another. This ensures that programs created by crossover are also trees and are syntactically valid. Figure 1(b) shows an example of a crossover operation.

Genetic Logic Programming, [33, 34] is a framework for inducing logic programs with evolutionary algorithms. A logic program is represented as a number of trees. The crossover operation is similar to crossover in genetic programming.

5 Nervous Systems

The information reservoir of an organism comprises not solely of that held in the genotype, but also includes information stored in other ways. For example, information is stored in the firing properties and topology of nerve cells in the nervous system. Changes to nerve cell properties allow the organism's behaviour to modify and adapt during its lifetime.

Evolutionary adaptation and nervous system adaptation operate at different time scales. Evolutionary learning does not influence the organism within its lifetime and therefore does not permit sufficiently fast adaptation to react to a dynamic environment. Consequently, changes to the organism's environment can only be responded to slowly, requiring of the order of generations not milliseconds. The nervous system, on the other hand, is one of a set of mechanisms that enable rapid adaptive behaviour. Learning in the nervous system therefore allows plasticity in the behaviour of an organisms, while genetically determined behaviour (or instinct) tends to be brittle.

5.1 Evolving Neural Networks

Evolutionary learning systems have been constructed that model an evolving nervous system. The nervous system is modelled by a neural network whose structure (topology) and weights correspond to interconnections between neurons and their firing properties. Neural architectures are designed by an evolutionary algorithm, similar to those described in previous sections. The genotype may encode information about neural network topology, the functions computed by the neuron (e.g. threshold, sigmoid, etc.), and the connection weights, [2]. Over several generations, the population evolves towards genotypes that correspond to high-fitness neural networks. This relation between evolutionary and neural network algorithms is illustrated in Figure 2.

6 Concepts and Symbolic Reasoning

In the 1920s, Wolgang Köhler studied learning and problem solving in chimpanzees. In a typical experiment, Köhler placed a chimpanzee in an enclosed area with a desirable piece of fruit, often a banana, out of reach. The animal had to use nearby objects (e.g. stack several boxes strewn around the enclosure) in order to obtain the fruit. Several interesting observations were made: (1) rather than being a gradual trial-and-error process, the solution was found suddenly; (2) once solved, repeating the problem resulted in few irrelevant moves; and (3) the animal was able to transfer what it had learned to novel situations, [1].

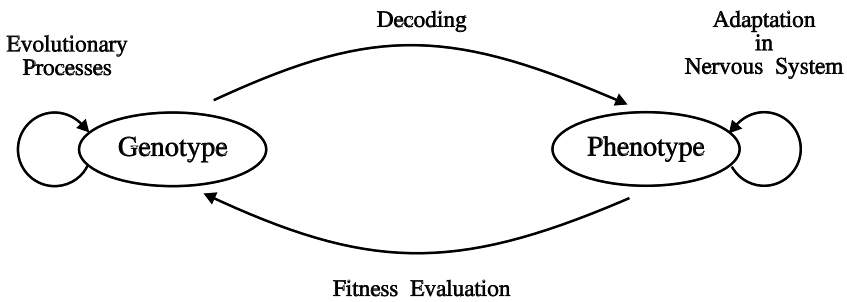

Figure 2: Evolutionary design of neural networks (adapted from [2]). The genotype is decoded to determine properties of a neural network. The neural network, interacting with the environment, adapts to it; and the better adapted, the greater the probability that the genotype will contribute to the population of the next generation.

But how networks of neurons achieve this kind of complex behaviour is still poorly understood and there remain many challenging problems for those trying to unravel the mysteries of the brain. One popular view is that the animal constructs an internal representation or model of the environment and the objects in it. The model might be a map of the environment or an abstract concept. It is then possible to reason about this model rather than operate on the world itself.

Objects in the environment are grouped into sets of properties or *concepts*. Treating different objects as members of the same concept with the same properties allows different environmental situations to be responded to in a uniform way, thus significantly reducing the complexity of the environment. Concepts may be combined to form propositions, and propositions may be combined to form further propositions. This combining process may be referred to as *reasoning*.

We now describe evolutionary learning systems that attempt to construct and manipulate symbolic internal models based on interaction with an environment.

6.1 Evolving Symbolic World Models

Logic, the rigorous treatment of reasoning, has been the focus of much study, originating with Aristotle's *Prior Analytics*. Later it was developed into a formal mathematical science in the 19th and 20th century primarily by Boole and Frege. Logic provides a formal language with which to describe and manipulate symbolic models of an environment. It is also a representation that is easily intelligible for humans.

Nevertheless, even though a logic may have well-defined properties, the conclusions drawn from an internal model may not coincide with the world it describes. It is therefore necessary for the model to accurately reflect the world, but the key problem is constructing this model. In the rest of this section we survey evolutionary approaches to constructing symbolic world models. We categorise approaches by their representation language, the search biases employed, and how background knowledge supplied by a domain expert is exploited.

Representation Language. A key strength of the genetic algorithm is the *implicit parallelism* in the information that is processed. Superficially, genetic algorithms appear only to process the binary strings in the population. However, Holland, [20], demonstrated that the genetic algorithm in fact processes a large amount of information concerning unseen hyperplanes. The significance of this is elaborated by Schaffer, [28]. "Since there are very many more than N hyperplanes represented in a population of N strings, this constitutes the only known example of the combinatorial explosion working to advantage instead of disadvantage". However, the larger the alphabet for each position in the string, the smaller the number of hyperplanes processed simultaneously. Consequently, a binary representation allows the optimal degree of implicit parallelism.

The logic-based approaches to evolutionary learning may be categorised into two classes depending on whether they try to exploit the implicit parallelism to an optimum. Approaches that attempt to exploit the implicit parallelism associate a logical representation with a binary string. The evolved binary strings are mapped to a logicals expressions and evaluated. Examples of these systems include learning classifier systems, [29, 19], COGIN, [16], GABIL, [6], REGAL, [13]. An alternative approach is to abandon the binary representation and the associated optimal implicit parallelism, and to operate on the logical representation directly. Such learning systems include SIA, [32], GIL, [23], Genetic Logic Programming, [33, 34], and EVIL1, [27].

Search bias. As the representation language becomes more general and expressive, the space that must be searched increases combinatorially. In order to restrict the solution space, mechanisms (or biases) are used to govern the candidate solutions considered. We briefly cover syntactic, semantic and preferences biases.

A syntactic bias may ensure that the candidate solutions constructed are syntactically valid. For example, when a binary string representation is used, the number of possible strings frequently does not coincide with the number of possible logical expressions. Consequently, some strings may have no mapping. An elementary syntactic bias would filter out these illegal strings. See for example GABIL, [6]. Another approach is to use a representation which can only form legal representations, e.g. Genetic Logic Programming, [33, 34].

The well-defined semantics of first order logic permits semantic properties of logical expressions to be described and used. Semantic biases may be used to ensure that a candidate solution is logically complete and consistent with respect to the training examples and that the solution is itself internally consistent. The use of such logical properties enables the use of generalisation and specialisation operations as found in SIA, [32], REGAL, [13], GIL, [23], and EVIL1, [27].

A further bias is a preference towards simpler candidate solutions.

Background knowledge. Most approaches to evolutionary learning begin *tabula rasa*, with an initial population of solutions generated randomly. Even though there may be expert domain knowledge available, the search space comprises of many unnecessary candidate solutions. There are essentially two approaches to bringing background knowledge to bear on a problem.

The first approach is to bias or *seed* the initial population of candidate solutions with background information. Consequently, the chances of a solution containing the background knowledge are increased. Examples include DOGMA, [17], which places user-supplied background rules into the initial population. In Genetic Logic Programming, [33, 34], the initial population includes rules constructed from training examples using the FOIL inductive logic programming algorithm, [26].

The other approach is to limit the candidate solutions only to those that are logically consistent with rules and facts supplied by the domain expert, as in EVIL1, [27].

We have presented a classification of evolutionary approaches to learning symbolic world models. We draw attention to the fact that there are no evolutionary learning systems that evolve neural networks capable of the kind of complex behaviour that may be achieved with symbolic representations. Instead, symbolic world models are manipulated directly, eliminating the intermediate neural network representation. The use of first-order logic as the representation language offers several advantages including comprehensible theories, deductive inference, and easily incorporated background knowledge. In the next section we consider language and its role in evolutionary learning.

7 Language

Recall from Section 5 that the genotype does not undergo changes within the lifetime of an organism and that a secondary representation, embodied in the nervous system, has evolved that permits more rapid adaptation. Consequently, the information accumulated *during* an organism's lifetime is not genetically propagated to subsequent generations. This learned behaviour is lost at the end of the lifetime of the organism. However, if an organism were capable of passing on this learned behaviour, or, more generally, were capable of acquiring experience from another organism, it may well have a selective advantage over other organisms that do not have this capacity. Indeed, mechanisms *have* evolved that allow information acquired during the lifetime of an organism to be propagated.

One such mechanism is the Baldwin effect. In 1896, Baldwin [3] proposed that the ability of an individual to learn can guide the evolutionary process. Furthermore, abilities that initially require learning are eventually replaced by the evolution of genetically determined mechanisms that do not require learning. Consequently, there is a gradual transferral of learned behaviours into the genotype. However, this process is slow, requiring of the order of generations for the learned behaviour to become genetically determined.

Another mechanism that avoids the loss of information acquired during the lifetime of an organism is language. A language provides a common framework for expressing the combination of symbols to describe complex notions such as an organism's environment. It permits the communication of information in a highly structured way and allows the exchange of information about symbolic world models. Language therefore provides a mechanism to avoid losing learned information at the end of an organism's lifetime. This description is necessarily oversimplistic, however, our aim is to highlight role of language as a mechanism for communicating symbolic world models.

In Section 9.2 it is demonstrated that communication in multi-agent search can significantly improve search performance over a non-cooperative search. The genetic algorithm has also been considered from the point of view of cooperative search, [4]. The population is viewed as a set of cooperating agents, and a generation is one of many repeated encounters among agents. During crossover, agents communicate parts of their solution (or schemata) to other agents. Test cases were presented that suggest that the genetic algorithm may at times yield a behaviour similar to a cooperative search.

There exist evolutionary learning systems that model agent communication in the form of recombination, as described in Section 4.1. There exist also languages for agent communication, such as KQML, [7], and frameworks for communicating inductive inferences in multi-agent learning systems, [5]. Yet no descriptions of evolutionary learning systems that employ language to implement communication have been found in the literature.

8 The Tinkerer's Evolving Toolbox[2]

The previous sections have described several milestones in evolutionary history and surveyed processes that have been abstracted to construct evolutionary learning algorithms. These algorithms all implement the processes of reproduction, selection, and competition and differ primarily in the underlying representation and the operators that produce variation.

The types of variation, listed in Table 1, progress from chance mutation to increasingly complex adaptive mechanisms. Indeed, there appears to be a trend: *from elementary beginnings, mechanisms have developed that permit increasingly complex adaptations to the environment through modifications to their morphology, physiology and behaviour.*

Variation Mechanisms
mutation
mutation + recombination
mutation + recombination + nervous system
mutation + recombination + nervous system + reasoning
mutation + recombination + nervous system + reasoning + language

Table 1: Variation Mechanisms in Evolutionary Adaptation

The work in evolutionary algorithms has reached the point where mutation, recombination, and nervous systems have been modelled. However, the literature reports no progress in developing the next step: evolving neural networks capable of reasoning. It is perhaps significant to point out that the evolution of reasoning from the nervous system took a very long time, even by evolutionary timescales. Therefore, it is perhaps unrealistic to expect to evolve reasoning in neural networks when time and computational resources are limited.

It appears that the variation mechanisms are implementing the following two processes: (1) the capability to adapt rapidly; and (2) allowing information to be steadily passed from generation to generation. If this is the case, then directly evolving symbolic world models may prove a suitable replacement for further complexities, such as neural networks and the Baldwin effect.

9 An Analysis

In order to better understand the success of learning systems based on natural evolution, it would be invaluable to comprehend the role and significance of the evolutionary processes described in Section 2. We attempt to do this by drawing on analyses of computational models of evolutionary learning. There is of course no guarantee that computational models accurately reflect biological

[2]The title of an article by Moxon and Thaler [25]

processes. Nevertheless, we hold that some useful abstraction must have been made as many of these models successfully tackle real problems.

9.1 Reproduction, Selection, and Competition

Reproduction, selection, and competition together play the role of determining which organisms will contribute to the next generation. As such, these processes may be viewed as resource allocation mechanisms that implement the following policy: those genotypes that fare best in the environment have a greater chance of constructing the next population.

Analysis of genetic algorithms, [20, 14], show the efficiency of this resource allocation. If we assume a part of an encoding (or schemata), k, of an individual is attributed a fitness that remains consistently above the average over several generations, then the number of instances of k will obey a geometric progression, and will exhibit exponential growth in frequency. That is, reproduction, selection and competition can propagate exponentially increasing and decreasing instances of a piece of information.

9.2 Communication and Cooperative Search

It may be observed that different mechanisms have evolved that facilitate the interaction of biological entities allowing information to be exchanged. Examples include genetic recombination, the interaction of neurons, and, at a higher level, human language. This repeated occurrence is a curious phenomenon and begs investigation of the significance of communication. To do this we examine communication from the point of view of multi-agent search.

The performance characteristics of interactive processes engaged in cooperative problem solving has been examined by Huberman, [21]. It was shown that for a wide class of problems, there is a highly nonlinear increase in performance due to the interactions between agents.

As a specific example, consider the problem of searching a d-dimensional vector, each of whose components can take b different values. The search problem involves finding a goal state among the b^d possible states.

However, suppose n agents cooperate on the solution of this problem, and that the problem is completed by the first agent to find the solution. Agents can exchange information, or *hints*, regarding the likely location of the goal state within the space. The effect of good hints is to move the goal state towards the beginning of the sequence of states examined by an agent. However, hints may not always be good and so can lead to an increase in the states that need to be examined.

Providing certain assumptions are observed, Huberman showed that the speed with which an agent finds solutions is distributed according to a log-normal distribution, similar to the one shown in Figure 3. This distribution is highly asymmetric with a long tail, signifying an enormous range of performance among the individual agents. This extended tail describes the improvement in performance due to the increased likelihood of having a few high performers when agents exchange hints.

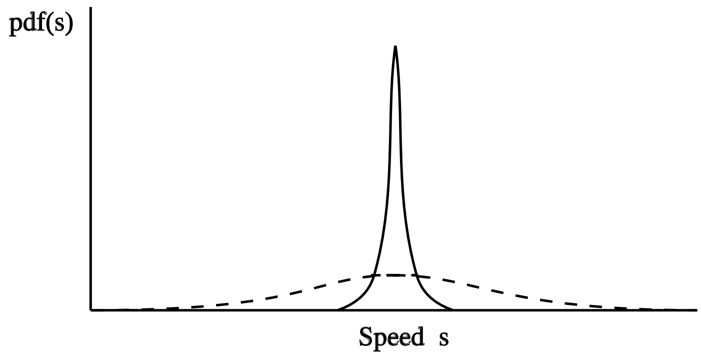

Figure 3: The distribution of agents according to their search performance S. The solid curved represents the case where agents do not communicate, while the dashed curve represents the interacting case. The curve is asymmetric since S cannot be less than zero, yet reaches very high values with non-zero probability. It is this long tail in the interacting case that indicates the improvement in performance. (adapted from [21]).

The performance of top performers increases in a highly nonlinear, multiplicative way as the population is composed of more diverse agents and as agents are more cooperative. This implies that the performance of a collection of cooperating agents *on average* increases through the exchange of hints[3].

This theoretical result provides a candidate explanation for the frequent evolution of communication between organisms.

10 Discussion

Through evolutionary history, mechanisms have evolved that permit increasingly rapid and complex adaptations to the environment. In this survey we have presented a selection of evolutionary learning systems characterising them by the mechanisms that have been abstracted, and in particular drawing attention to their relative positions on the timeline of natural evolution. A set of necessary processes was identified, which comprise of reproduction, selection, competition, and a source of variation. As we progress down the evolutionary timeline the mechanisms for causing variation become increasingly complex.

[3]Further results quantifying the increase in performance may be found in [21].

It was shown that agent communication leads to highly nonlinear increases in agent performance and more complex forms of agent communication, such as language, promises to be a fruitful area for further investigation.

11 Acknowledgements

The author would like to thank Nathalie Santré, John Hunt, Guillem Bernat and Alan Lush for useful discussions and comments on earlier drafts. This work is supported by a college grant from the University of Wales, Aberystwyth.

References

[1] R. L. Atkinson, R.G. Atkinson, E.E. Smith, and D.J. Bem. *Introduction to Psychology.* Harcourt Brace College Publishers, 1993. 11th edition.

[2] Karthik Balakrishnan and Vasant Honovar. Evolutionary design of neural architectures – a preliminary guide to the literature. Technical Report CS TR#95-01, Artificial Intelligence Group, Iowa State University, January 1995.

[3] J. M. Baldwin. A new factor in evolution. *American Naturalist*, 30:441–451, 1896.

[4] Helen G. Cobb. Is the genetic algorithm a cooperative learner? In *Proceedings of the Workshop on the Foundations of Genetic Algorithms and Classifier Systems*, pages 277–296. Morgan Kaufmann, July 1992.

[5] W.H.E. Davies and P. Edwards. The communication of inductive inferences. In *Lecture Notes in Artifical Intelligence (1221): Distributed Artificial Intelligence Meets Machine Learning: Learning in Multi-Agent Environments*, pages 223–241. Springer Verlag, Berlin, 1997.

[6] Kenneth A. DeJong, William M. Spears, and Diana F. Gordon. Using genetic algorithms for concept learning. *Machine Learning*, 13:161–188, 1993.

[7] Tim Finin, Rich Fritzon, Don McKay, and Robin McEntire. KQML – A language and protocol for knowledge and information exchange. In *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence*, pages 126–136, Seatle, WA, July 1994.

[8] David B. Fogel. *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence.* IEEE Press, New York, 1995.

[9] D.B. Fogel. Applying evolutionary programming to selected travelling salesman problems. *Cybernetics and Systems*, 63:111–114, 1993.

[10] D.B Fogel. Evolutionary programming: an introduction and some current directions. *Statistics and Computing*, 4:113–129, 1994.

[11] L.J. Fogel. Autonomous automata. *Industrial Research*, 4:14–19, 1962.

[12] L.J. Fogel. *On the Organisation of the Intellect.* PhD dissertation, UCLA, 1964.

[13] A. Giordana and F. Neri. Search-intensive concept induction. *Evolutionary Computation*, 3(4):375–416, 1995.

[14] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, Reading, MA, 1989.

[15] David E. Goldberg. Genetic and evolutionary algorithms come of age. *Communications of the ACM*, Vol. 37:113–119, March 1994.

[16] David Perry Greene and Stephen F. Smith. Competition-based induction of decision models from examples. *Machine Learning*, 13:229–257, 1993.

[17] Jukka Hekanaho. Background knowledge in GA-based concept learning. Technical Report TUCS-TR-2, TUCS - Turku Centre for Computer Science, April 10 1996.

[18] A. Hoffman. *Arguments on Evolution: A Paleontologist's Perspective.* Allen and Unwin, London, 1989.

[19] J. H. Holland. Escaping brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems. In T. Mitchell, R. Michalski, and J. Carbonell, editors, *Machine Learning, Volume 2*, chapter 20, pages 593–623. Morgan Kaufmann, San Mateo, CA, 1986.

[20] John H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.

[21] Bernardo A. Huberman. The performance of cooperative processes. *Physica D*, 42:38–47, 1990.

[22] J. S. Huxley. The evolutionary process. In J. Huxley, A.C. Hardy, and E.B. Ford, editors, *Evolution as a Process*, pages 9–33. Collier Books, New York, 1963.

[23] Cezary Z. Janikow. A knowledge-intensive genetic algorithm for supervised learning. *Machine Learning*, 13:189–228, 1993.

[24] John R. Koza. *Genetic Programming: On the Programming of Computers by Natural Selection.* MIT Press, Cambridge, MA, USA, 1992.

[25] E. Richard Moxon and David S. Thaler. The tinkerer's evolving toolbox. *Nature*, 387:659–662, 12 June 1997.

[26] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5(3):239–266, August 1990.

[27] Philip Reiser. *EVIL1*: a learning system to evolve logical theories. In *Proc. Workshop on Logic Programming and Multi-Agent Systems (International Conference on Logic Programming)*, pages 28–34, July 1997.

[28] J. D. Schaffer. Some effects of selection procedures on hyperplane sampling by genetic algorithms. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing.* Pittman, 1987.

[29] Stephen F. Smith. *A Learning System Based on Genetic Adaptive Algorithms.* PhD thesis, University of Pittsburgh, 1980.

[30] Paul D. Sniegowski, Philip J. Gerrish, and Richard E. Lenski. Evolution of high mutation rates in experimental populations of E. coli. *Nature*, 387:703–705, 12 June 1997.

[31] F. Taddei, M. Radman, J. Maynard-Smith, B. Toupance, P.H. Gouyon, and B. Godelle. Role of mutator alleles in adaptive mutation. *Nature*, 387:700–702, 12 June 1997.

[32] Gilles Venturini. SIA: a supervised inductive algorithm with genetic search for learning attributes based concepts. In *Proceedings of the European Conference on Machine Learning*, pages 280–296. Springer Verlag, 1993.

[33] Man Leung Wong and Kwong Sak Leung. Inductive logic programming using genetic algorithms. In J.W. Brahan and G.E. Lasker, editors, *Advances in Artificial Intelligence – Theory and Application II*, pages 119–124, 1994.

[34] Man Leung Wong and Kwong Sak Leung. The genetic logic programming system. *IEEE Expert Magazine: Intelligent Systems and their Applications*, 10(2):68–76, October 1995.

[35] D.E. Wooldridge. *The Mechanical Man: The Physical Basis of Intelligent Life.* McGraw-Hill, New York, 1968.